

Open Research Online

The Open University's repository of research publications and other research outputs

A Bottom-Up Approach for Licences Classification and Selection

Conference or Workshop Item

How to cite:

Daga, Enrico; d'Aquin, Mathieu; Motta, Enrico and Gangemi, Aldo (2015). A Bottom-Up Approach for Licences Classification and Selection. In: 2015 Workshop on Legal Domain And Semantic Web Applications (LeDA-SWAn 2015), 1 Jun 2015, Portoroz, Slovenia.

For guidance on citations see [FAQs](#).

© 2015 The Authors



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Version of Record

Link(s) to article on publisher's website:

<http://cs.unibo.it/ledaswan2015/papers/daga-et-al-ledaswan2015.pdf>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

A Bottom-Up Approach for Licences Classification and Selection

Enrico Daga¹, Mathieu d'Aquin¹, Enrico Motta¹, and Aldo Gangemi²

¹ Knowledge Media Institute, The Open University
Walton Hall, Milton Keynes, United Kingdom
{enrico.daga,mathieu.daquin,enrico.motta}@open.ac.uk
<http://kmi.open.ac.uk>

² Université Paris13, Sorbonne Cité
CNRS UMR7030, France, and
Institute of Cognitive Sciences and Technologies - CNR
Via S. Martino della Battaglia 44, 00185 Rome (RM), Italy
aldo.gangemi@{univ-paris13.fr,cnr.it}
<http://www.univ-paris13.fr>
<http://istc.cnr.it>

Abstract. Licences are a crucial aspect of the information publishing process in the web of (linked) data. Recent work on modeling of policies with semantic web languages (RDF, ODRL) gives the opportunity to formally describe licences and reason upon them. However, choosing the right licence is still challenging. Particularly, understanding the number of features - permissions, prohibitions and obligations - constitute a steep learning process for the data provider, who has to check them individually and compare the licences in order to pick the one that better fits her needs. The objective of the work presented in this paper is to reduce the effort required for licence selection. We argue that an ontology of licences, organized by their relevant features, can help providing support to the user. Developing an ontology with a bottom-up approach based on Formal Concept Analysis, we show how the process of licence selection can be simplified significantly and reduced to answering an average of three/five key questions.

Keywords: RDF, Licences and Linked Data, Formal Concept Analysis

1 Introduction

Licence specification is an important part of the data publishing process on the web. Recently, a part of the Semantic Web and Linked Data community has been focusing on providing support to the expression of policies on the semantic web. The Open Digital Rights Language (ODRL) provides an ontology for representing policies in the semantic web, and it is used and extended to formally express permissions, prohibitions and duties that licences include³. The

³ <http://www.w3.org/ns/odrl/2/ODRL21>

RDF Licenses database ⁴ is a first notable attempt at developing a knowledge base of licences described following ODRL. However, identifying suitable licences is still not a trivial task for a data publisher. In the current version, ODRL identifies more than fifty possible actions to be used as permissions, prohibitions or obligations, and there are ontologies that extend ODRL adding even more fine grained policies (e.g. LDR⁵). Therefore, not only are there many licences that can be applied, but each might include any subset of the many possible features (permitted, prohibited and required actions), that need to be explored in order to obtain a small selection of comparable licences to choose from.

The question that this paper aims to answer is: *How can we reduce the effort for licence identification and selection?* We advance the hypothesis that an ontology defining relevant classes of licences, formed on the basis of the *key features* of the instances, should facilitate the selection and identification of a suitable licence. The methodology applied relies on a bottom-up approach to ontology construction based on Formal Concept Analysis (FCA). We developed a tool, *Contento*, with the purpose of analysing data about licences using FCA, in order to generate a concept lattice. This concept lattice is used as a draft taxonomy of licence classes that, once properly annotated and pruned, can be exported as an OWL ontology and curated with existing ontology editors. We applied this approach to the use case of licence identification, and created a service to support data providers in licence selection by asking a few key questions about their requirements. We show that, with this service, we can reduce the selection of licences from comparing more than fifty possible licence features, to answering on average three to five questions.

The next section surveys related work. Section 3 describes the process of building the ontology, the *Contento* tool and the modeling choices that have been made. In Section 4 we report on the application of the ontology in a service for identification of suitable licences for data providers. Ultimately, we discuss some future work in the concluding Section 5.

2 Related work

Licence recommendation is very common on the web, particularly for software. Services like <http://choosealicense.com/> are usually based on common and well known concerns, and recommend a restricted number of trusted solutions. The Creative Commons Choose service⁶ shares with our approach a workflow based on few questions. However, it is an ad-hoc tool which focuses on selecting a Creative Commons licence. Differently, we are interested in applying a knowledge-based approach, where the way information about licences and requirements is modelled guides the path to the solution.

⁴ <http://datahub.io/dataset/rdflicense>

⁵ <http://oeg-dev.dia.fi.upm.es/licensius/static/ldr/>

⁶ <https://creativecommons.org/choose/>

The Open Digital Rights Language (ODRL) is a rights expression language formalised as an XML Schema⁷. Recently, an alternative representation based on RDF/OWL has been identified as the backbone for representing policies in the semantic web [1]. The RDF Licenses database [2] includes the description of licences in RDF⁸. We used this database as starting point for the present work. However, population and curation of such knowledge base is clearly a necessary step for licence recommendation systems. For example, the descriptions do not specify the types of assets a licence is eligible for (and we don't cover this aspect in the present paper). The enrichment of the possible terms to express policies will contribute to increase the precision and quality of the descriptions (see LiMO⁹, L4LOD¹⁰ and ODRS¹¹). Applying natural language processing techniques, like the ones proposed in [3], can facilitate the process of data acquisition.

Licentia [4] is a tool for supporting users in choosing a licence for the web of data. Similarly to our approach, it is based on the RDF licence database. The user selects possible permissions, obligations and duties extracted from the licence descriptions, in order to specify her requirements. The system applies reasoning over the databases of licences, proposing a list of compatible ones to choose from. With this approach the user needs to perform an action for each of its requirements. Our approach restricts the number of questions through the inferences implied by the classification of licences in a hierarchy (e.g.: any “share alike” licence allows distribution) and only suggests the ones for which a solution actually exists.

The approach proposed in this paper relies on an ontology of licences as a means for licence selection. Such an ontology has been created following a bottom-up approach. Bottom-up approaches for ontology design have been commonly applied in knowledge engineering [5] and we use here one particular method based on Formal Concept Analysis (FCA) [6]. FCA has been successfully used in the context of recommender systems [7, 8]. Moreover, FCA has been proposed in the past to support ontology design and other ontology engineering tasks [9, 10]. In the present work we use FCA as a learning technique to boost the early stage of the ontology design.

3 Building the ontology

Our hypothesis is that an ontology can help on orienting the user in the complex set of existing licences and policies. The RDF Licenses database contains 139 licences expressed in RDF/ODRL. Our idea is therefore to start from the data to create the ontology. The reason for choosing a bottom-up approach to ontology construction is also that the data will include only policies that are relevant.

⁷ <http://www.w3.org/TR/odrl/>

⁸ <http://purl.org/NET/rdflicense>

⁹ <http://data.opendataday.it/LiMo>

¹⁰ <http://ns.inria.fr/l4lod/v2/l4lodv2.htm>

¹¹ <http://schema.theodi.org/odrs/>

In order to support the production of the ontology we implemented a bottom-up ontology construction tool called *Contento*, which relies on FCA. FCA has the capability of classifying collections of objects depending on their *features*. The input of a FCA algorithm is a *formal context* - being a binary matrix having the full set of objects as rows and the full set of attributes as columns. Objects and attributes are analysed and clustered in *closed concepts* by FCA. In FCA, a concept consists of a pair of sets - objects and attributes: the objects being the extent of the concept and the attributes its intent.

For the mathematical definition, FCA introduces the derivation operator $'$. For a set of objects X , we define X' as the set of attributes all shared by the objects in X . Similarly, for a set of attributes Y , Y' is the set of objects that share all attributes in Y . A *closed concept* is a pair of objects and attributes (X, Y) so that $X' = Y$ and $X = Y'$. It is possible to derive a close concept (also called *formal concept*) from a set of objects using a simple routine:

1. Select a set of objects X .
2. Derive the set of attributes X' .
3. Derive in the same way the related objects $(X')'$.
4. (X'', X') is a close concept.

The same process can be performed starting from a set of attributes. A subsumption relation can be established between formal concepts in order to define an order on the set of formal concepts in a formal context. As a result, formal concepts are organized in a hierarchy, starting from a top concept (e.g., *Any*), including all objects and an empty set of attributes, towards a bottom concept (e.g., *None*), with an empty set of objects. Moreover, this ordered set forms a mathematical structure: the concept *lattice*.

The objective of the *Contento* tool is to support the user in the generation and curation of concept lattices from formal contexts (binary matrixes) and to use them as drafts of semantic web ontologies.

3.1 Contento

*Contento*¹² has been developed to create, populate and curate FCA formal contexts and associated lattices, also interpreted as taxonomies of concepts. Formal contexts can be created and populated from scratch. Sets of items can be managed with a number of features in the *Collections* section. The user can assign the role of objects' set and attributes' set to two collections, thus to generate a formal context. Figure 1 presents the formal context browser of *Contento*. Each context is represented as a list of relations between one object and one attribute and a *hold* status: *yes*, *no* or *undefined*. The *undefined* status has been included to indicate that the relation has not been supervised yet. The user can then incrementally populate the formal context by choosing whether each object/attribute association occurs or not. This can be done conveniently thanks to a set of filtering options that can reduce the list to only a subset of the context to be analysed. Data can be filtered in different ways:

¹² <http://bit.ly/contento-tool>

- by object name (or all that have a given attribute)
- by attribute name (or all that have a given object)
- by status (holds, does not hold, to be decided)

Relations of context #69: "RDF License 8"				
Object	<input type="text" value="http://purl.org/NET/rdfliblicense/cc-by"/>	Attribute	<input type="text" value="Any"/>	Holds
have attr	<input type="text" value="Any"/>	have obj	<input type="text" value="Any"/>	<input type="text" value="Any"/>
				<input type="button" value="Filter (113)"/>
Relation	Objects	Attributes	Holds	
#45525	#2189 http://purl.org/NET/rdfliblicense/cc-by4.0	#2163 permission http://www.w3.org/ns/odrl/2/read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#45573	#2189 http://purl.org/NET/rdfliblicense/cc-by4.0	#2211 permission http://www.w3.org/ns/odrl/2/write	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#45619	#2189 http://purl.org/NET/rdfliblicense/cc-by4.0	#2212 permission http://www.w3.org/ns/odrl/2/derive	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#45669	#2189 http://purl.org/NET/rdfliblicense/cc-by4.0	#2213 permission http://www.w3.org/ns/odrl/2/append	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fig. 1. Contento: formal context browser and editor. In this example, we have fixed the object in order to review its relations with the attributes.

Therefore, the user can display only the relations that need to be checked (the ones with status *undefined*), focus on the extent of a specific attribute or on the intent of an object. Moreover, she can display the set of relations having for object any that include a specific attribute (or vice versa). Eventually, the user can set all filtered relations to a given state in bulk, if meaningful. With this interface, the binary matrix can be incrementally populated to constitute a proper input for a FCA algorithm.

In many cases, however, a ready made binary matrix can be imported from pre-existing data. In this case the formal context is created directly from that, ready to be used to generate the concept lattice with the procedure provided by Contento.

Contento implements the Chein algorithm [11] to compute concept lattices. The result of the algorithm is stored as a *taxonomy*. A taxonomy can be navigated as an ordered list of concepts, from the top to the bottom, each of them including the extent, the intent and links to upper and lower concept bounds in the hierarchy (see Figure 2). In addition, the tool shows which objects and attributes are *proper* to the concept, i.e. do not exist in any of the upper (for attributes) or lower (for objects) concepts.

Moreover, it can be visualized and explored as a concept lattice (Figure 3). The lattice can be navigated by clicking on the nodes. Focusing on a single node, the respective upper and lower branches are highlighted, to facilitate the navigation to the user. Similarly, objects and attributes from the focused node can be selected, thus highlighting all nodes in the hierarchy sharing all of the selected features (in orange in Figure 3). Contento supports the user on the curation of the concept hierarchy, to transform it from a concept lattice to a draft ontology taxonomy, through the annotation of each concept with a label and a comment, and the pruning of unwanted concepts. This last operation implies an

Concept #7874 Save Delete in lattice Close

↑ Any

Name
Re

Comment
Should the licence permit the reproduction (copy) of the data?

Intent
 permission <http://creativecommons.org/ns#Reproduction>
 permission <http://purl.oclc.org/NET/ldr/ns#Reproduction>
 permission <http://www.editeur.org/onix-pl/extract-char>
 permission <http://www.editeur.org/onix-pl/extract-page>
 permission <http://www.editeur.org/onix-pl/extract-word>
 permission <http://www.w3.org/ns/odrl/2/copy>
 11 proper | 0 inherited | 0 highlighted

Extent
<http://purl.org/NET/rdflib/GNU-LGPL3.0>
<http://purl.org/NET/rdflib/W3C1.0>
<http://purl.org/NET/rdflib/APACHE2.0>
<http://purl.org/NET/rdflib/ARTISTIC2.0>
<http://purl.org/NET/rdflib/BOOST1.0>
<http://purl.org/NET/rdflib/cc-by-nc-nd3.0>
 2 proper | 40 inherited | 0 highlighted

↓ #SRC ↓ #NC ↓ #Sell ↓ #Pre ↓ #D ↓ #A

Fig. 2. Contento: each concept is presented showing the extent, the intent and links to upper and lower concept bounds in the hierarchy. The portion of the intent not included in any of the upper concepts (called *proper*) is highlighted, as well as any objects not appearing in lower concepts. Concepts can be annotated and deleted.

adjustment of the hierarchy, by building links between lower and upper bounds of the deleted node (only if no other path to the counterpart exists). As a result, relevant concepts can be qualified, and concepts that are not relevant for the task at end can be removed.

Taxonomies can be translated into OWL ontologies. The user can decide how to represent the taxonomy in RDF, what terms to use to link concepts, objects and attributes, and whether items need to be represented as URIs or literals. Ultimately, these export configurations can be shared and reused. For example, Contento offers a default profile, using example terms, or a SKOS profile.

3.2 The ontology

For the use case at hand, we used Contento to support the creation of the Licence Picker Ontology (LiPiO)¹³, starting from data in the RDF Licences database. The data has been preprocessed in order to produce a binary matrix to be imported in Contento. The preprocessing included reasoning on SKOS-like relations between ODRL actions¹⁴. Moreover, we reduced the number of licences from the initial 139 to 48 by removing localized versions (for instance Creative Commons CC-BY-SA 3.0 Portugal). In this case, the licences are the objects of the matrix, while the set of attributes represent the policies, expressed as ODRL

¹³ <http://bit.ly/licence-picker-ontology>

¹⁴ We also introduced some changes in the original descriptions, that will be contributed to evolve the RDF Licence database itself.

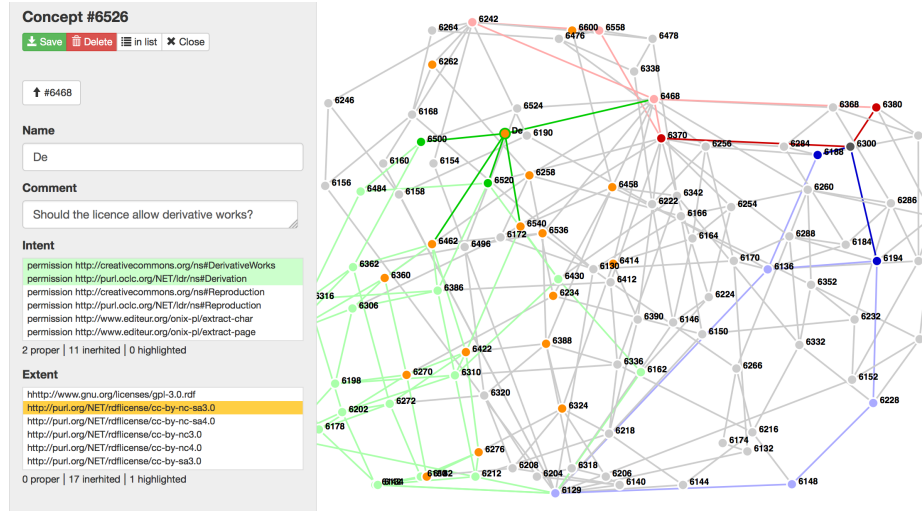


Fig. 3. Contento: the lattice explorer for annotation and pruning. The branching of the current concept is presented in the lattice in green (on the left side of the picture). The user can still point to other nodes to inspect the branching of other concepts (on the right side of the picture, the lower branch being displayed in blue and the upper in red). By selecting one or more items in the extent or intent of the concept, all the nodes sharing the same are bolded in orange.

permissions, prohibitions or duties. Below is an example taken from the input CSV:

```
http://purl.org/NET/rdflicense/cc-by4.0,permission http://www.w3.org/ns/odrl/2/copy,1
http://purl.org/NET/rdflicense/allrightsreserved,prohibition http://www.w3.org/ns/odrl/2/copy,1
http://purl.org/NET/rdflicense/MOZILLA2.0,duty http://www.w3.org/ns/odrl/2/shareAlike,0
```

In the above excerpt, the “CC-BY” licence permits to *copy*, the “All rights reserved” policy prohibits it, and the “Mozilla 2.0” licence does not include a *share-alike* requirement.

The CSV has been imported in the Contento tool that created the formal context automatically. After that, a concept lattice was generated. The lattice included 103 concepts organized in a hierarchy, the top concept representing *All* the licences, while the bottom concept, *None*, includes all the attributes, and no licence. Figure 3 shows the lattice as it looked like at this stage of the process. In this phase, the objective is to inspect the concepts and, for each one of them, to perform one of the following actions:

- If the concept is meaningful, name it and annotate it with a relevant question (e.g. “should others be allowed to distribute the work?”) in the comment field;
- If the concept is not meaningful or useful, it can be deleted (with the lattice being automatically adjusted).

We judged the meaningfulness of a concept by observing its intent (set of features). If the concept was introducing new features with respect to the upper concepts, then it is kept in the lattice, given a name and annotated with a question. In the case its intent does not include new features (it is a union of the intents of the respective upper concepts), then it is deleted, because the respective licences will necessarily be present in (at least one of) the upper concepts, and no new question need to be asked to identify them. With this process the lattice has been reduced significantly, and proper names and questions have been attached to the remaining concepts (almost 20% of the initial lattice). Figure 4 displays

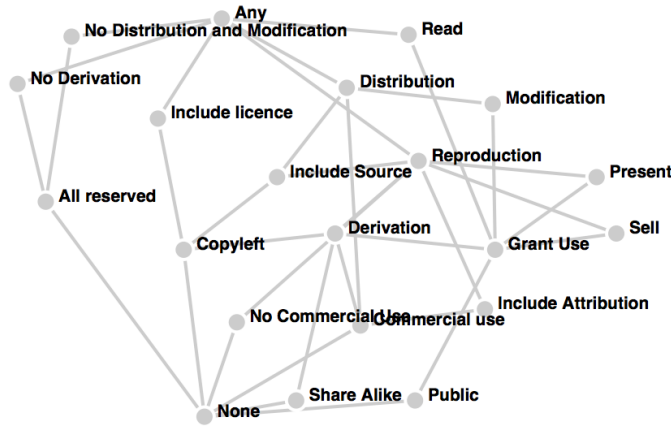


Fig. 4. Contento: the annotated and pruned concept lattice.

the resulting lattice, labels being synthetic names referring to policies/attributes that have been introduced in that point of the hierarchy; i.e. according to the *key features* that define the concept in relation to its parents.

The resulting annotated taxonomy has been exported as OWL ontology as the initial draft of the the Licence Picker Ontology. The draft included a sound hierarchy of concepts. Both concepts (classes) and licences were annotated with the respective set of policies. Because the policies were expressed as plain literal on a generic *has* property (the data being manipulated as object/attribute pairs by the FCA based tool), a small refactoring permitted to reintroduce the RDF based descriptions with ODRL. The Licence Picker ontology¹⁵ currently contains 21 classes linked to 45 licences with a *is-A* relation. Each class is associated with a relevant question to be asked that makes explicit the key feature of the included set of licences. The ontology embeds annotations on the classes about the policies included in all the licences of a given concept, and a ODRL based description of permissions, prohibitions and duties of each instance.

¹⁵ <http://bit.ly/licence-picker-ontology>

4 Pick the licence

The Licence Picker Ontology has been designed to support data providers in choosing the right policy under which to publish their data. In order to evaluate this ontology we applied it in a service for licence selection. The Licence Picker Webapp is an ontology driven web application¹⁶. The user is engaged in answering questions regarding her requirements to reach a small set of suitable licences to compare, like in the following guide example. We consider a scenario, inspired

Questions (6)

Should the licence prohibit the distribution and modification of the data?

Should the licence permit to read (access) the data?

Should the licence prohibit derivative works?

Should the licence permit the reproduction (copy) of the data?

Should the licence permit the distribution/sharing of the data?

Should the licence require to include copyright and license notices?

Licences (45)

All rights reserved

cc:DerivativeWorks cc:Distribution cc:Reproduction id:odc:Derivation id:odc:Distribution id:odc:Reproduction id:databaseRight id:extraction id:IPRRight id:reuseUtilization
 on:extract-char on:extract-page on:extract-word od:aggregate od:annotate od:anonymize od:append od:appendTo od:archive od:concurrentUse od:copy
 od:derive od:digitize od:display od:distribute od:execute od:export od:extract od:extractChar od:extractPage od:extractWord od:grantUse od:index
 od:install od:license od:modify od:move od:play od:present od:print od:read od:reproduce od:sell od:share od:textToSpeech od:transform
 od:translate od:use od:write od:writeTo

Public domain

cc:DerivativeWorks cc:Distribution cc:Reproduction id:odc:Derivation id:odc:Distribution id:odc:Reproduction id:databaseRight id:extraction id:IPRRight id:reuseUtilization
 on:extract-char on:extract-page on:extract-word od:aggregate od:annotate od:anonymize od:append od:appendTo od:archive od:concurrentUse od:copy
 od:derive od:digitize od:display od:distribute od:execute od:export od:extract od:extractChar od:extractPage od:extractWord od:grantUse od:index
 od:install od:license od:modify od:move od:play od:present od:print od:read od:reproduce od:sell od:share od:textToSpeech od:transform
 od:translate od:use od:write od:writeTo

Fig. 5. Licence Picker Webapp: the user is engaged in answering questions.

from our work on smart cities data hubs [12], in which sensors are installed in a city to detect how busy different areas are at different times, as information to be provided to local retailers, restaurants, etc. This information is collected in a data store and offers access to statistics through a number of web-based services. The managers of the data store needs to choose a license to attach to the data in order to limit their exploitation to the expected uses. They want (a) the data to be accessible and copied for analysis, but (b) to not be modified or redistributed to third parties. In addition, (c) commercial uses should be allowed, but (d) the data consumers should attribute the source of the data to the owner of the data store.

The Licence Picker Webapp welcomes the user with forty-five possible licences and a first set of questions, as show in Figure 5. One of them catches the eye of the user: *Should the licence prohibit derivative works?* She promptly answers *Yes*. The set of possible licences is reduced to five, and the system propose a single question: *Should the licence prohibit any kind of use (All rights reserved)?* This time the user answers *No*, because they want the users to use

¹⁶ <http://bit.ly/licence-picker-webapp>

the information to boost the activities in the data store. As a result, the system proposes to pick one of four licences. The user notices that all of them require an attribution statement and prohibit to produce derivative works. Two of them also prohibit the use for commercial purposes, so the user decides to choose the *Creative Commons CC-BY-ND 4.0* licence.

The example above shows an important property of the approach presented in the paper. As the licences are classified by the mean of their features, and the classes organized in a hierarchy, we can notably reduce the number of actions to be taken to obtain a short list of comparable licences. The user had four requirements to fulfill, more than fifty existed in the database, and she could get an easy comparable number of licences with only two steps.

5 Conclusions and future work

Licences are an important part of the data publishing process, and choosing the right licence may be challenging. By applying the Licence Picker Ontology (LiPiO), this task is reduced to answering an average of three to five questions (five being the height of the class taxonomy in LiPiO) and assessing the best licence from a small set of choices. We showed how our approach reduces significantly the effort of selecting licences in contrast with approaches based on feature exploration. In addition, a bottom-up approach on ontology building in this scenario opens new interesting challenges. The RDF description of licences is an ongoing work, modeling issues are not entirely solved and we expect the data to evolve in time, including eventually new licences and new types of policies. For example, in our use case the data has been curated in advance in order to obtain an harmonized knowledge base, ready to be bridged to the *Contento* tool. This clearly impacts the ontology construction process and the application relying on it, as different data will lead to different classes and questions. This gives the opportunity to explore methods to automate some of the curation tasks (especially pruning) and to integrate changes in the formal context incrementally, to support the ontology designer in the adaptation of the ontology to the changes performed in the source knowledge base. Such evolutions do not impact the Licence Picker Webapp, because changes in the ontology will be automatically reflected in the tool. We foresee that the description of licences will be extended including other relevant properties - like the type of assets a licence can be applied to. The advantage of the proposed methodology is that it can be applied to any kind of licence feature, not only policies.

The *Contento* tool was designed to support the task at the center of the present work. However, the software itself is domain independent, and we plan to apply the same approach to other domains. Ultimately, we want to compare *Contento* to other similar tools, for example *ToscanaJ* [13], and perform a user based evaluation.

References

1. Simon Steyskal and Axel Polleres. Defining expressive access policies for linked data using the ODRL ontology 2.0. In Harald Sack, et al., editors, *Proceedings of the 10th International Conference on Semantic Systems (SEMANTiCS 2014)*, pages 20–23. ACM, 2014.
2. Víctor Rodríguez-Doncel, Serena Villata, and Asunción Gómez-Pérez. A dataset of RDF licenses. In Rinke Hoekstra, editor, *Legal Knowledge and Information Systems. JURIX 2014: The Twenty-Seventh Annual Conference*. IOS Press, 2014.
3. Elena Cabrio, Alessio Palmero Aprosio, and Serena Villata. These are your rights. In Valentina Presutti, et al., editors, *The Semantic Web: Trends and Challenges*, volume 8465 of *Lecture Notes in Computer Science*, pages 255–269. Springer International Publishing, 2014.
4. Cristian Cardellino, Serena Villata, Fabien Gandon, et al. Licentia: a tool for supporting users in data licensing on the web of data. In Matthew Horridge, Marco Rospocher, and Jacco van Ossensbruggen, editors, *Proceedings of the ISWC 2014 Posters Demonstrations Track, a track within the 13th International Semantic Web Conference (ISWC 2014)*, Riva del Garda, Italy, 21 October 2014.
5. Paul E Van Der Vet and Nicolaas JI Mars. Bottom-up construction of ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 10(4):513–526, 1998.
6. Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.
7. Patrick du Boucher-Ryan and Derek Bridge. Collaborative recommending using formal concept analysis. *Knowledge-Based Systems*, 19(5):309–315, 2006.
8. Xiaohui Li and Tomohiro Murata. A knowledge-based recommendation model utilizing formal concept analysis and association. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 4, pages 221–226. IEEE, 2010.
9. Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In *Concept Lattices*, pages 189–207. Springer, 2004.
10. Marek Obitko, Vaclav Snasel, Jan Smid, and V Snasel. Ontology design with formal concept analysis. In *CLA*, volume 110, 2004.
11. Michel Chein. Algorithme de recherche des sous-matrices premieres dune matrice. *Bull. Math. Soc. Sci. Math. RS Roumanie*, 13(61):21–25, 1969.
12. Mathieu d’Aquin, Alessandro Adamou, Enrico Daga, et al. Dealing with diversity in a smart-city datahub. In Tope Omitola, John Breslin, and Payam Barnaghi, editors, *Proceedings of the Fifth Workshop on Semantics for Smarter Cities, a Workshop at the 13th International Semantic Web Conference (ISWC 2014)*, Riva del Garda, Italy, 19 October 2014. CEUR-WS.org.
13. Peter Becker and Joachim Hereth Correia. The ToscanaJ Suite for implementing conceptual information systems. In *Formal Concept Analysis*, pages 324–348. Springer, 2005.